

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
In re Application of
JOAN LLACH-PINSACH ET AL

Atty. Docket

PHF 99,593

Serial No. 09/610,712

Group Art Unit: 2771

Filed: JULY 6, 2000

Examiner

Title: AUTOMATIC EXTRACTION METHOD OF THE STRUCTURE OF A VIDEO
SEQUENCE
Honorable Commissioner of Patents and Trademarks
Washington, D.C. 20231

CLAIM FOR PRIORITY

Sir:

A certified copy of the European Application No.
99401684.8 filed JULY 6, 1999 referred to in the Declaration of the
above-identified application is attached herewith.

Applicants claim the benefit of the filing date of said
European application.

Respectfully submitted,

By

Russell Gross
Russell Gross, Reg. 40,007
Attorney
(914) 333-9631

Enclosure

CERTIFICATE OF MAILING

It is hereby certified that this correspondence is being deposited with the
United States Postal Service as first-class mail in an envelope addressed to:
COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

On

By

MY22GRH0.SW0



TECH CENTER 2700

OCT -6 2000

RECEIVED

THIS PAGE BLANK (USPTO)



Europäisches
Patentamt

European
Patent Office

Office européen
des brevets



Bescheinigung

Certificate

Attestation

Die angehefteten Unterla-
gen stimmen mit der
ursprünglich eingereichten
Fassung der auf dem näch-
sten Blatt bezeichneten
europäischen Patentanmel-
dung überein.

The attached documents
are exact copies of the
European patent application
described on the following
page, as originally filed.

Les documents fixés à
cette attestation sont
conformes à la version
initialement déposée de
la demande de brevet
européen spécifiée à la
page suivante.

RECEIVED
OCT - 6 2000
TECH CENTER 2700

Patentanmeldung Nr. Patent application No. Demande de brevet n°

99401684.8

**CERTIFIED COPY OF
PRIORITY DOCUMENT**

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

I.L.C. HATTEN-HECKMAN

DEN HAAG, DEN
THE HAGUE,
LA HAYE, LE

29/06/00

THIS PAGE BLANK (USPTO)

RECEIVED
JAN 10 1970



Europäisches
Patentamt

European
Patent Office

Office européen
des brevets

Blatt 2 der Bescheinigung
Sheet 2 of the certificate
Page 2 de l'attestation

Anmeldung Nr.:
Application no.:
Demande n°: 99401684.8

Anmeldetag:
Date of filing: 06/07/99
Date de dépôt:

Anmelder:
Applicant(s):
Demandeur(s):
Koninklijke Philips Electronics N.V.
5621 BA Eindhoven
NETHERLANDS

Bezeichnung der Erfindung:
Title of the invention:
Titre de l'invention:

Analysis of video sequences: table of contents and index creation

In Anspruch genommene Priorität(en) / Priority(ies) claimed / Priorité(s) revendiquée(s)

Staat:
State:
Pays:

Tag:
Date:
Date:

Aktenzeichen:
File no.
Numéro de dépôt:

Internationale Patentklassifikation:
International Patent classification:
Classification internationale des brevets:

/

Am Anmeldetag benannte Vertragsstaaten:
Contracting states designated at date of filing: AT/BE/CH/CY/DE/DK/ES/FI/FR/GB/GR/IE/IT/LI/LU/MC/NL/PT/SE
Etats contractants désignés lors du dépôt:

Bemerkungen:
Remarks:
Remarques:

THIS PAGE BLANK (USPTO)

ANALYSIS OF VIDEO SEQUENCES: TABLE OF CONTENTS AND INDEX CREATION

Joan Llach, Philippe Salembier

Universitat Politècnica de Catalunya
Campus Nord - Mòdul D5
C/ Gran Capità, s/n
08034 Barcelona, Spain
E-mail: {jllach, philippe}@gps.tsc.upc.es

ABSTRACT

This paper deals with the representation of video sequences useful for tasks such as long-term analysis, indexing or browsing. A *Table Of Content* and *index* creation algorithm is presented, as well as additional tools involved in their creation. The proposed method does not assume any *a priori* knowledge about the content or the structure of the video sequence. It is therefore a generic technique. Some examples are presented in order to assess the performance of the algorithm.

1. INTRODUCTION

In the framework of video analysis for indexing application (for example application related to the MPEG-7 standard), the representation of video sequences is an important issue. It is not enough to describe the content of a video sequence but also to develop techniques which are able to automatically create these descriptions. In this paper, a new technique that automatically creates *TOCs* and *indexes* is presented. The proposed algorithm only relies on visual information, unlike other techniques which use both video and audio information [5].

The goal of the *TOC* is to define the structure of the video sequence in a hierarchical fashion, like in a text document. The original sequence should be subdivided in sub-sequences which can also be divided in shorter sub-sequences, etc. At the end of this division process, the shortest entity to be described is the *micro-segment*. The *index* is also a hierarchical structure but it does not describe the structure of the sequence but the occurrences of similar content.

The creation of the *TOC* and *index* follows the same strategy. The whole process is divided into three parts: shot detection, segmentation of shots and shot clustering. The first step splits the sequence into shots, which are the input data for the next steps. The second step divides each shot into *micro-segments* which are sub-components of a shot where the camera activity is homogeneous. These *micro-segments* constitute the lowest level of the *TOC* or *index* representations. Finally, the third step of the algorithm creates the hierarchical structures by clustering the detected video shots.

This paper is structured as follows. Section 2 deals with the shot detection algorithm. The temporal segmentation of shots is detailed in section 3, whereas their clustering is explained in section 4. Finally, some conclusions are driven in section 5.

2. SHOT DETECTION

The first step in the *TOC* and *index* creation splits the sequence into shots. Taking into account that a shot is a set of contiguous frames without editing effects, the algorithm has to detect the transitions between consecutive shots. These transitions can be abrupt or more sophisticated, like dissolves, fades, etc. The shot detection algorithm consists of two main steps: computation of the *mean Displaced Frame Difference mDFD* curve and its segmentation into shots.

2.1. Computation of the *mDFD* curve

The *mDFD* curve is obtained using a variation of the *Displaced Frame Difference* into account both luminance and chrominance. $\{I_x, I_y\}_{t=Y,U,V}$ denotes the luminance (*Y*)

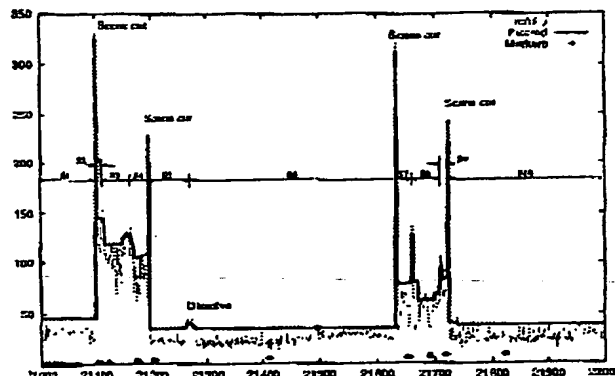


Figure 1: *mDFD* for the *news2* sequence (frame 21 000 to 22 000, thin line). The filtered curve (thick line) and the detected markers (horizontal segments) are also plotted. S1-S10 depict the limits of the detected shots.

and the chrominance (*U, V*) components of frame at time *t*, the *DFD* is given by $DFD_k(i, j; t-1, t+1) = f_k(i, j, t+1) - f_k(i - d_x(i, j), j - d_y(i, j), t-1)$ and the *mDFD* by:

$$mDFD(t) = \frac{1}{I_x I_y} \sum_k^{Y,U,V} w_k \sum_{i,j}^{I_x, I_y} |DFD_k(i, j; t-1, t+1)| \quad (1)$$

where I_x and I_y are the image dimensions and w 's are the weights for *Y, U, V* components. In the example shown in Fig. 1 the weights have been set to $\{w_Y, w_U, w_V\} = \{1, 3, 3\}$, which are typical values. The highest peaks of the curve correspond to the abrupt transitions (frames 21 100, 21 195, 21 633 and 21 724). On the other side, the oscillation from frame 21 260 to frame 21 279 corresponds to a dissolve. The presence of large moving foreground objects in frames 21 100-21 195 and 21 633-21 724 creates high level oscillations of the *mDFD* curve.

2.2. Segmentation of the *mDFD* curve

The second block of the shot detection algorithm detects the video editing effects. Our approach has been to adapt classical spatial segmentation strategies to the case of temporal segmentation. Most of classical shot detection techniques use threshold-based segmentation to extract the highest peaks of the *mDFD* or another type of mono-dimensional curve [7]. Although a large number of shots can actually be detected by this approach, this class of algorithms is not robust and quite sensitive to noise. It is in particular difficult to detect peaks of small contrast corresponding to fading or special effects. The solution that we have used is a homogeneity-based approach relying on morphological tools. The *mDFD* curve is processed following four steps:

1. *Simplification by temporal filtering.* Morphological closing with a 1D structuring element of length l_{min} . With this operation, negative peaks the length of which is less than l_{min} frames are removed. Physically, the l_{min} parameter defines the duration of the shortest shot to be detected.

2. *Simplification by positive contrast filter.* The effect of the filter is to remove positive peaks that have a positive contrast lower than the parameter c .
3. *Marker extraction.* Each marker corresponds to the kernel of one shot. So, each marker must cover a portion of the curve with a high probability to belong to a single shot. Because contiguous frames that belong to the same shot are quite similar, the value of the $mDFD$ will be small around those frames. Thus, to extract the markers a negative contrast filter is used because it detects the relative minimums of the curve. The same c parameter than in the previous step is used in the negative contrast filter.
4. *Watershed.* This is the final step and its purpose is to propagate the markers on the curve until all points are assigned to a marker. The propagation process is performed by applying a watershed algorithm [8] on the $mDFD$ curve using as initial markers those obtained in the previous step.

Figure 1 shows the filtered curve (both temporal filter and positive and negative contrast filter), the resulting markers and detected shots using $l_{min} = 10$ and $c = 10$. Even though some over-segmentation appears around frames 21 150 and 21 700, both the scene cuts and the dissolve have been correctly detected. It is important to mention that the over-segmentation is not a problem because the next steps of the algorithm eliminate this effect.

3. TEMPORAL SEGMENTATION OF VIDEO SHOTS USING CAMERA MOTION PARAMETERS

The goal of the temporal segmentation algorithm is to split each shot into several micro-segments which present a high level of homogeneity on the camera motion parameters. The data used to perform the segmentation are the camera motion parameters currently used in MPEG-7 [2]. This algorithm is applied to each shot separately and consists of two steps: 1) each shot is over segmented into several micro-segments which must present a perfect homogeneity (see section 3.1), and 2) a merging process is applied while the homogeneity level of the set of micro-segments is below a predefined threshold.

In order to segment the shot, it is necessary to define a distance to compare sub-segments and a parameter which allows the assessment of the quality of a micro-segment or a partition (i.e., set of micro-segments). In both cases, we use a motion histogram, defined as follows:

$$H_s[i] = \frac{N_i}{L_s}, i \in \{\text{PanLeft, PanRight, ZoomIn, ZoomOut, Fix, ...}\} \quad (2)$$

where s represents the label of the segment inside the shot, i the motion type, L_s the length of segment s and N_i the number of frames of segment s with motion type i . Note that each bin of the histogram H shows the percentage of frames with a specific type of motion. $\sum H_s[i]$ may be higher than 1.0, because different motions can appear concurrently.

3.1. Homogeneity

We assume that a segment is perfectly homogeneous when it presents a single combination of camera motion parameters along all its frames. A segment is not homogeneous when it presents important variations on these parameters. The segment homogeneity is computed on its histogram (Eq. 2). If a segment is perfectly homogeneous, the histogram bins are equal to either 1.0 or 0.0. If a segment is not perfectly homogeneous, the bins can present intermediate values.

To measure the segment homogeneity, we measure how much its histogram differs from the ideal one. The distance corresponding to bins with high values is the difference between the bin value and 1.0. Analogously, for bins with small values, the distance is the bin value itself. Mathematically, the homogeneity of a segment s is given by:

$$H(s) = \sum_i \min(H_s[i], 1 - H_s[i]) = \begin{cases} 1.0 - H_s[i], & \text{if } H_s[i] \geq 0.5 \\ H_s[i], & \text{if } H_s[i] < 0.5 \end{cases} \quad (3)$$

where H_s is the histogram of the segment s and i indicates the different motion types. The homogeneity of a shot S is equal to the homogeneity of its segments weighting by the length of each of them, as shown in the following equation:

$$H(S) = \frac{1}{L_S} \sum_{j=1}^N L_j \cdot H(s_j) \quad (4)$$

where $L_S = \sum_{j=1}^N L_j$ is the total length of shot S and N the number of segments it contains. Note that small values of H correspond to high levels of homogeneity.

The distance between two segments, s_1 and s_2 , is the homogeneity of the segments union.

$$d(s_1, s_2) = H(s_1 \cup s_2) \quad (5)$$

3.2. Algorithm

The temporal segmentation algorithm consists of the following steps:

1. *Initial over-segmentation.* In this step, the shot is over-segmented in order to obtain a set of perfectly homogeneous micro-segments. Mathematically, the following condition must be fulfilled:

$$H(s) = 0, \forall s \in S$$

2. *Fusion order.* In this step, the distance between all neighboring segments (temporally connected) is computed using Eq. 5. The closest pair of segments is then selected for possible merging in the next step.
3. *Fusion criterion.* To decide if the selected pair of segments are going to be merged, we compute the homogeneity of the shot (Eq. 4) assuming that the minimum distance segments have already been merged and then the following criterion is applied:

$$\begin{cases} \text{Merge,} & \text{if } H(S) \leq \Theta_H \\ \text{Do not merge,} & \text{if } H(S) > \Theta_H \end{cases}$$

Note that the fusion criterion is global: the decision depends on the homogeneity of the resulting partition and not — exclusively — on the homogeneity of the resulting segment. If the merging is done, a new iteration starts at the second step of the algorithm. The merging process ends when there is no pair of neighboring segments that can be merged.

The algorithm allows different levels of detail as shown in the example of Fig. 2.

4. SHOT CLUSTERING

The shot clustering process is divided into two parts: shot merging and tree structuring. In the first step, pairs of shots are grouped together creating a binary tree. In the second step, the binary tree is restructured in order to reflect the similarity present in the video sequence.

4.1. Shot merging

The shot merging algorithm yields a binary tree which represents the merging order of the initial shots (see section 2). In this tree, the leaves represent the initial shots, the top node represents the whole sequence and the intermediate nodes represent sequences that are created by the merging of several shots. The merging criterion is defined by a distance between shots, merging first the closest shots. In order to compute the distance between shots it is necessary to define a shot model that provides the features to be compared, and to set the neighborhood links between them, which indicate what merging can be done. The *TOC* and *index* creation only differ in the neighborhood links. The former sets a link between each pair of temporally connected shots, the latter between all pair of shots.

The process ends when all the initial shots have been merged into a single node or when the minimum distance between all couples

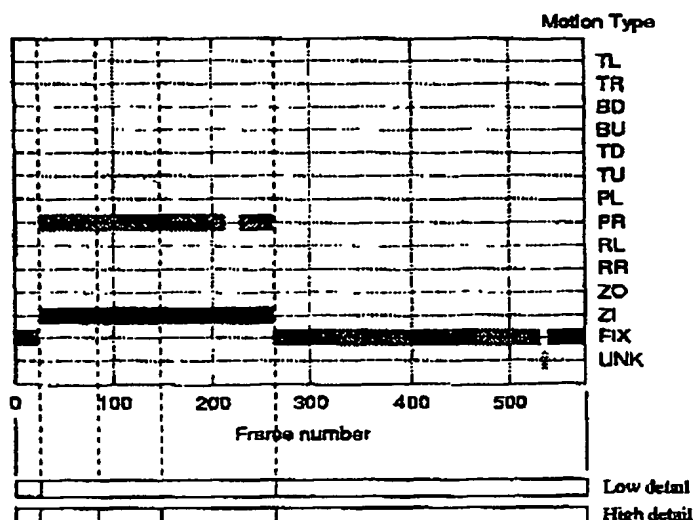


Figure 2: Example of temporal segmentation of shot #8 of MPEG-7 nhkvideo sequence. Above: Camera motion parameters chronogram; below: Two segmentations using different levels of homogeneity.

of linked nodes is greater than the specified threshold [10]. In the latter case, not a single tree but a set of binary trees—a forest of binary trees—is obtained.

4.1.1. Shot and sequence model

The shot model must allow us to compare the content of several shots to decide what shots must be merged and which is their merging order. In still images, luminance and chrominance are the main features of the image [9, 3]. In a video sequence, due to the temporal evolution, motion is an important source of information [3, 6, 4]. So, average images, histograms of luminance/chrominance information (YUV components) and motion information (x and y components of motion vectors) are used to model the shots.

4.1.2. Merging distance

In order to compute the distance between a pair of nodes, three different cases must be considered depending on the type of nodes. At the beginning of the merging process, the nodes represent a single shot: the average of the Euclidean distance between the different components of the model is used. In the following steps of the process, the distance must be computed between nodes that may represent more than a single shot: the minimum and maximum distance between all pair of shots (one from each node) is used [1].

4.1.3. Merging algorithm

The merging algorithm consists of three preliminary steps and the merging process itself. In the first step, nodes are modeled according to the model previously described. In the second step, neighborhood links between shots are set and in the third step the distance between linked shots is computed.

Since the whole algorithm must be able to create both the TOC and the index of a video sequence, the criterion used to set the neighborhood links differs depending on the main objective. In the TOC creation process, neighborhood links are set according to temporal connectivity. So, a pair of shots will be linked and therefore may be merged only if they are consecutive in the time domain. On the other hand, all pair of shots will be linked and therefore may be merged in the index creation process.

Once the algorithm has been initialized, the merging process starts performing the following steps:

- **Get minimum distance link.** Both the minimum and the maximum distance are computed for every pair of linked nodes. Between the minimum distance, the maximum distance is

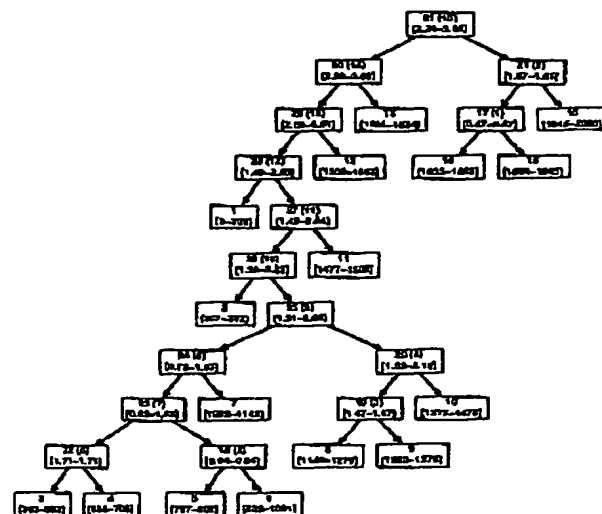


Figure 4: Binary tree created by the shot merging algorithm. The initial shots are shown in Fig. 3. The top node represents the whole sequence, while the leaf nodes represent the initial shots.

checked. If maximum distance is higher than the maximum distance threshold d_{max} , then the link is discarded. Otherwise, the link is taken into account. Once all links have been scanned, the minimum distance link is obtained.

- **Check distance criterion.** In order to decide if the nodes pointed by the minimum distance link must be merged, the minimum distance is compared to the minimum distance threshold d_{min} . If the minimum distance is higher than the threshold, no merging is performed and the process ends. Otherwise, pointed nodes are merged and the process goes on.
- **Merge nodes.** Nodes pointed by the minimum distance links are merged.
- **Update links.** This steps update those links to take into account the merging that has been done.
- **Update distances.** Once links have been updated, the distance of those links which point to the new node is recomputed.
- **Check top node.** This step checks the number of remaining nodes. If all initial shots have been merged into a single node, the process ends. Otherwise, a new iteration begins.

The merging process may yield a single tree if all the initial shots are similar enough or a forest if initial shots are quite different.

An example of binary tree for TOC creation is shown in Fig. 4. Inside the leaf nodes of this tree, we have indicated its label, and between brackets [...] the starting and ending frame number of the shot. Inside the remaining nodes, we have indicated its label, between parenthesis (...) the fusion order and between brackets the minimum and maximum distance between its two siblings.

4.2. Tree structuring

The purpose of the this algorithm is to restructure the binary tree into an arbitrary tree that should reflect more clearly the video structure. To this end, nodes that have been created by the binary merging process but that do not convey any relevant information should be removed. The criterion used to decide if a node must appear in the final tree is based on the variation of the similarity degree (distance) between the shots included in the node:

- If the analyzed node is the root node (or one of the root nodes, if various binary trees have been obtained after the merging algorithm), then the node should be preserved in the final tree.
- If the analyzed node is a leaf node (i.e., it corresponds to initial shot), then it has also to remain in the final tree.

SO (S) 394d 7H101

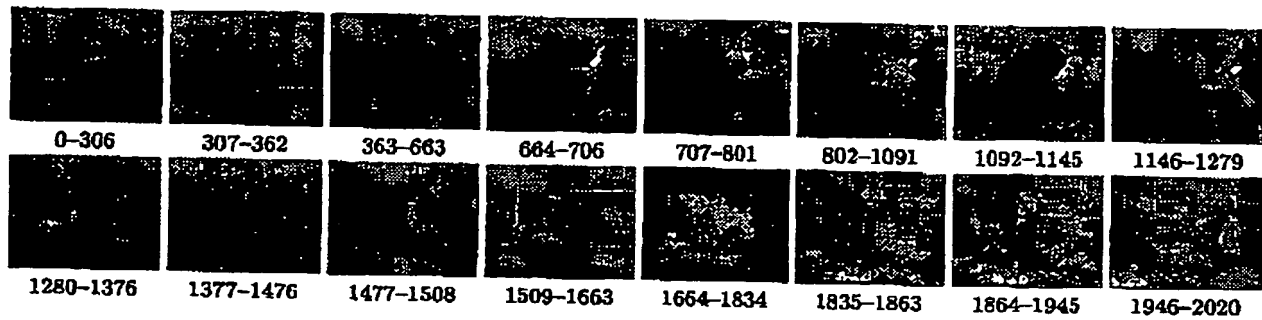


Figure 3: Detected shots for drama sequence, using $l_{min} = 20$ and $c = 10$. Each image represents the central frame of the shot, the numbers show the starting and ending frame of the shot. Shots are ordered from left to right and from top to bottom.

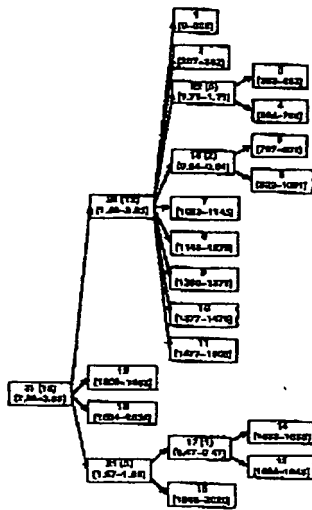


Figure 5: Tree yielded by the tree structuring algorithm.

- Otherwise, the node will be kept in the final tree if the following conditions are satisfied:

$$\begin{cases} |d_{min}(\text{analyzed node}) - d_{min}(\text{parent node})| < \Theta \\ |d_{max}(\text{analyzed node}) - d_{max}(\text{parent node})| < \Theta \end{cases}$$

As shown in Fig. 5, the tree resulting from the restructuring step represents more clearly the structure of the video sequence. Nodes in the first level of the hierarchy (28,12,13,21) represents the four scenes of the sequence, while nodes in the third —or occasionally fourth— represent the initial shots.

5. CONCLUSIONS

In this paper, techniques for *TOC* and *index* creation has been presented, video shot detection and *micro-segment* segmentation have been presented. The shot detection algorithm is able to cope with abrupt transitions as well as smooth ones even though it over-segments those portions of the video sequence with a high level of motion. In order to solve this drawback, a confidence mask will be added to the motion vectors to discard those parts of the image which can not be properly estimated when computing the *mDFD*.

The ability of the *micro-segment* segmentation algorithm to compute partitions at different level of detail allows to adjust the granularity of the resulting *TOC* or *index* to take into account different applications. Moreover, the algorithm is quite robust in front of mistakes in the camera motion parameters.

Finally, the *TOC* and *index* creation algorithm provides an unified method to compute both structures. Moreover, new pieces of information can easily be introduced in the merging process. For example, the algorithm has been developed to deal with the case of knowing before hand some characteristics of the sequence (e.g. the type of video), this information can

be taken into account to improve the performance of the algorithm in specific scenarios.

6. REFERENCES

- [1] D. A. Adjeroh, M. C. Lee and I. King, "A Distance Measure for Video Sequence Similarity Matching", in *Proceedings of the 1998 International Workshop on Multimedia Database Management Systems*, pp. 72-79, June 1998.
- [2] R. Jasinschi, A. She, T. Naveen, A. Tabatabai, S. Jeannin and B. Mory, "Motion Descriptors for Content Based Video Representation", to be published in *Image Communications: Special Issue on MPEG-7 proposals*, 1999.
- [3] Rakesh Mohan, "Video Sequence Matching", in *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'98)*, vol. 6, pp. 3697-3700, June 1998.
- [4] Jean-Marc Odobez and Patrick Bouthemy, "Direct Incremental Model-Based Image Motion Segmentation for Video Analysis", *Technical note no. 1129*, September 1997.
- [5] C. Saraceno and R. Leonardi, "Indexing Audio-Video Databases through a joint audio and video processing", to be published in *International Journal of Imaging Systems and Technology*.
- [6] Harpreet S. Sawhney and Serge Ayer, "Compact Representations of Videos Through Dominant and Multiple Motion Estimation", in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 814-830, August 1996.
- [7] Taehwan Shin, Jae-Gon Kim, Hankyu Lee and Jinwoong Kim, "Hierarchical Scene Change Detection in an MPEG-2 Compressed Video Sequence", in *Proceedings of the 1998 IEEE International Symposium on Circuits and Systems (ISCAS'98)*, vol. 4, pp. 253-256, March 1998.
- [8] L. Vincent and P. Soille, "Watersheds in digital spaces: and efficient algorithm based on immersion simulations", in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 39, pp. 1845-1855, December 1991.
- [9] Minerva M. Yeung and Bede Liu, "Efficient Matching and Clustering of Video Shots" in *Proceedings of 1995 IEEE International Conference on Image Processing (ICIP'95)*, vol. 1, pp. 338-341, October 1995.
- [10] Minerva M. Yeung and Boon-Lock Yeo, "Time-constrained Clustering for Segmentation of Video into Story Units", in *Proceedings of ICPR'96*, pp. 375-380, 1996.

CLAIMS :

1. A preprocessing method provided for defining, before tasks such as long-term analysis, indexing or browsing, the structure of a video sequence and describing it, said method comprising a first step, provided for creating a table of content, that gives
5 said structure in a hierarchical fashion, and indexes, that describe the occurrences of similar content.
2. A method according to Claim 1, in which the creation of the table of content and indexes is carried out on the basis of a strategy comprising the following sub-steps :
10 a first splitting sub-step for sub-dividing the sequence into shots, a second splitting sub-step for sub-dividing each shot into shorter entities called micro-segments homogeneous according to a predetermined criterion, and a third clustering sub-step for creating said hierarchical structure by clustering the detected video shots.
3. A method according to Claim 2, in which said first splitting sub-step
15 comprises the following operations : a computation of the mean displaced frame difference curve, obtained using a variation of the displaced frame difference to take into account both luminance and chrominance information, and a segmentation of said curve, based on the detection of video editing effects.
4. A method according to claim 3, in which said curve is processed following
20 four steps : a simplification by temporal filtering, a simplification by positive contrast filtering, a marker extraction, and a propagation of the markers until all points are assigned to a marker.
5. A method according to anyone of claims 2 to 4, in which each shot is splitted
into several micro-segments presenting a high level of homogeneity on the motion parameters of the camera by which said video sequence has been delivered.
- 25 6. A method according to anyone of claims 2 to 5, in which said third clustering sub-step is divided into a first shot merging operation, provided for creating a binary tree by grouping together pairs of shots, and a second tree structuring operation, provided for restructuring said tree in order to reflect the similarity present into the video sequence.

THIS PAGE BLANK (USPTO)

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

THIS PAGE BLANK (USPTO)